Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

Claim 1 (original): A method for encoding an instruction to save processor core register values, comprising:

encoding in a first field of the instruction whether a first value in a first register is to be saved at one of a first location or a second location in a stack memory;

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the first value in the stack memory at the first location having an address value equal to A plus a second value in a second register; and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value at the second location in the stack memory having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number.

Claim 2 (original): The method of claim 1, further comprising:

encoding in the first field of the instruction whether a third value in a third register is to be saved at one of a third location or a fourth location in the stack memory;

if, based on the encoding in the first field of the instruction, the third value is to be saved as a value associated with storage already allocated on the stack, saving the third value in the stack memory at the third location having an address value equal to C plus the second value in a second register; and

if, based on the encoding in the first field of the instruction, the third value is to be saved as a value associated with storage not yet allocated on the stack, saving the third value at the fourth location in the stack memory having an address value equal to D plus the second value in the second register, wherein only one of C and D is a positive number.

Claim 3 (original): The method of claim 2, further comprising:

encoding in the first field of the instruction whether a fourth value in a fourth register is to be saved at one of a fifth location or a sixth location in the stack memory; and

encoding in the first field of the instruction whether a fifth value in a fifth register is to be saved at one of a seventh location or an eighth location in the stack memory.

Claim 4 (original): The method of claim 1, further comprising:

encoding in a second field of the instruction a sixth value to be used to adjust the second value in the second register.

Claim 5 (original): The method of claim 4, further comprising:

encoding at least one least significant bit of the sixth value in a first 16-bit portion of the instruction; and

encoding at least one most significant bit of the sixth value in a second 16-bit portion of the instruction.

Claims 6-9 (cancelled)

Claim 12 (original): A method for saving processor core register values in memory, comprising:

processing an instruction having an encoded value in a first field of the instruction, using a processor core having a plurality of registers, wherein the instruction has been encoded by encoding in the first field of the instruction whether a first value in a first register is to be saved at one of a first location or a second location in a stack memory,

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the first value in the stack memory at the first location having an address value equal to A plus a second value in a second register, and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value at the second location in the stack memory having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number;

saving the first value in the stack memory at the first location only if based on the encoded value the first value in the first register is to be saved as a value associated with storage already allocated on the stack; and

saving the first value in the stack memory at the second location only if based on the encoded value the first value in the first register is to be saved as a value associated with storage not yet allocated on the stack.

Claim ~~11~~ 13 (original): The method of claim ~~10~~ 12, further comprising:

adjusting the second value in the second register based on a value encoded in a second field of the instruction.

Claims 12-13 (cancelled)

Claim 14 (original): A computer readable medium comprising a microprocessor core embodied in software, the microprocessor core comprising:

a plurality of registers;

means for processing a first instruction having an encoded value in a first field of the first instruction, wherein the first instruction has been encoded by encoding in the first field of the instruction whether a first value in a first register is to be saved at one of a first location or a second location in a stack memory,

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the first value in the stack memory at the first location having an address value equal to A plus a second value in a second register, and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value at the second location in the stack memory having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number;

means for saving the first value in the stack memory at the first location only if based on the encoded value the first value is to be saved as a value associated with storage already allocated on the stack; and

means for saving the first value in the stack memory at the second location only if based on the encoded value the first value is to be saved as a value associated with storage not yet allocated on the stack.

Claim 15 (original): The computer readable medium of claim 14, further comprising:

means for adjusting the second value in the second register based on a third value encoded in a second field of the instruction.

Claim 16 (original): The computer readable medium of claim 14, wherein the microprocessor core further comprises:

means for processing a second instruction having an encoded value in a first field of the second instruction, wherein the second instruction has been encoded by encoding in a first field of the second instruction whether a value stored in the stack memory is to be restored to the first register; and

means for restoring the value stored in the stack memory to the first register only if based on the encoded value in the first field of the second instruction the value stored in the stack memory is to be restored to the first register.

Claim 17 (original): A microprocessor core, comprising:

a plurality of registers;

means for processing a first instruction having an encoded value in a first field of the first instruction, wherein the first instruction has been encoded by encoding in the first field of the instruction whether a first value in a first register is to be saved at one of a first location or a second location in a stack memory,

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the first value in the stack memory at the first location having an address value equal to A plus a second value in a second register, and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value at the second location in the stack memory having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number;

means for saving the first value in the stack memory at the first location only if based on the encoded value the first value is to be saved as a value associated with storage already allocated on the stack; and

means for saving the first value in the stack memory at the second location only if based on the encoded value the first value is to be saved as a value associated with storage not yet allocated on the stack.

Claim 18 (original): The processor core of claim 17, further comprising:

means for adjusting the second value in the second register based on a third value encoded in a second field of the instruction.

Claim 19 (original): The processor core of claim 17, further comprising:

means for processing a second instruction having an encoded value in a first field of the second instruction, wherein the second instruction has been encoded by encoding in a first field of the second instruction whether a value stored in the stack memory is to be restored to the first register; and

means for restoring the value stored in the stack memory to the first register only if based on the encoded value in the first field of the second instruction the value stored in the stack memory is to be restored to the first register.

Claim 20 (original): A decoder for decoding instructions and providing control signals to an execution core, said decoder comprising:

means for decoding a first instruction, wherein the first instruction has been encoded by encoding in a first field of the instruction whether a first value in a first register is to be saved at one of a first location or a second location in a stack memory,

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the first value in the stack memory at the first location having an address value equal to A plus a second value in a second register, and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value at the second location in the stack memory having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number; and

means for decoding a second instruction, wherein the second instruction has been encoded by encoding in a first field of the second instruction whether a value in the stack memory at the second location, is to be restored to the first register.

Claim 21 (original): The decoder of claim 20, further comprising:

means for decoding a second field of the first instruction that contains a value used to adjust a value in the second register.

Claim 22 (original): The decoder of claim 20, further comprising:

means for decoding a second field of the second instruction that contains a value used to adjust a value in the second register.

Claim 23 (original): A mapper for mapping instructions, said mapper comprising:

means for mapping a first instruction, wherein the first instruction has been encoded by encoding in a first field of the instruction whether a first value in a first register is to be saved at one of a first location or a second location in a stack memory,

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the first value in the stack memory at the first location having an address value equal to A plus a second value in a second register, and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value at the second location in the stack memory having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number; and

means for mapping a second instruction, wherein the second instruction has been encoded by encoding in a first field of the second instruction whether a value in the stack memory at the second location, is to be restored to the first register.

Claim 24 (original): The mapper of claim 23, further comprising:

means for mapping a second field of the first instruction that contains a value used to adjust a value in the second register.

A

Claim 25 (original): The mapper of claim 23, further comprising:

means for mapping a second field of the second instruction that contains a value used to adjust a value in the second register.

Claims 26-27 (cancelled)

Claim ~~28~~ 26 (original): A method, comprising:

processing a first instruction having an encoded value in a first field of the instruction, the encoded value indicating whether a first value in a first register is to be saved at a first location or a second location in a stack memory;

saving the first value in the stack memory at the first location if, based on the encoded value, the first value is a value associated with storage not yet allocated on the stack; and

saving the first value in the stack memory at the second location if, based on the encoded value, the first value is a value associated with storage already allocated on the stack.

Claim ~~29~~ 27 (original): The method of claim ~~28~~ 26, wherein the first instruction includes a second field defining a stack frame, the first location being disposed within the stack frame and the second location being disposed outside the stack frame.

28 26
Claim 30 (original): The method of claim 28, further comprising:

processing a second instruction having an encoded value in a first field of the instruction, the encoded value indicating whether the first value stored in the stack memory may be restored to the first register.

29 28
Claim 31 (original): The method of claim 30, wherein the first value may be restored to the first register if the first field of the second instruction indicates the first value is a value associated with storage not yet allocated on the stack.

30
Claim 32 (original): A microprocessor system, comprising:

a stack memory; and

a processor, coupled to the memory, that services an instruction having an encoded value in a field of the instruction, wherein the instruction has been encoded by encoding in the field of the instruction whether a first value in a register is to be saved at one of a first location or a second location in the stack memory,

if, based on the encoding in the field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the value in the stack memory at the first location having an address value equal to A plus a second value in a second register, and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value in the stack memory at the second location having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number.

Claim ~~33~~ 6 (original): A computer readable medium comprising a microprocessor core embodied in software, the microprocessor core comprising:

a plurality of registers;

a decoder capable of decoding an instruction having an encoded value in a first field of the instruction, wherein the instruction has been encoded by encoding in the first field of the instruction whether a first value in a first register is to be saved at one of a first location or a second location in a stack memory,

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the first value in the stack memory at the first location having an address value equal to A plus a second value in a second register, and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value at the second location in the stack memory having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number; and

an execution unit capable of saving the first value in the stack memory at the first location if, based on the encoded value, the first value is to be saved as a value associated with storage already allocated on the stack, and capable of saving the first value in the stack memory at the second location if, based on the encoded value, the first value is to be saved as a value associated with storage not yet allocated on the stack.

Claim 7 (original): The computer readable medium of claim 6, wherein said decoder is capable of decoding a second field of the instruction that contains a third value used to adjust the second value in the second register, and wherein said execution unit is capable of adjusting the second value in the second register based on the third value.

Claim 8 (original): The computer readable medium of claim 6, further comprising:

a mapper capable of mapping the first instruction to a predetermined instruction width format configuration that is capable of being decoded by said decoder.

Claim 9 (original): A microprocessor core, comprising:

a plurality of registers;

a mapper capable of mapping an instruction to a predetermined instruction width format configuration, the instruction having an encoded value in a first field of the instruction, wherein the instruction has been encoded by encoding in the first field of the instruction whether a first value in a first register is to be saved at one of a first location or a second location in a stack memory,

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage already allocated on the stack, saving the first value in the stack memory at the first location having an address value equal to A plus a second value in a second register, and

if, based on the encoding in the first field of the instruction, the first value is to be saved as a value associated with storage not yet allocated on the stack, saving the first value at the second location in the stack memory having an address value equal to B plus the second value in the second register, wherein only one of A and B is a positive number; and

an execution unit capable of saving the first value in the stack memory at the first location if, based on the encoded value, the first value is to be saved as a value associated with storage already allocated on the stack, and capable of saving the first value in the stack memory at the second location if, based on the encoded value, the first value is to be saved as a value associated with storage not yet allocated on the stack.

Claim ~~37~~ 10 (original): The microprocessor core of claim ~~36~~ 9, further comprising:

a decoder capable of decoding the predetermined instruction width format configuration and generating execution unit control signals.

Claim ~~38~~ 11 (original): The microprocessor core of claim ~~36~~ 9, further comprising:

a cache for storing the instruction.